Nonthermal Module User's Manual

John C. Houck, houck@space.mit.edu Apr 1, 2010

Contents

1 Introduction **5**

2 Examples **7**

2.1 X-ray Synchrotron Emission 7

2.2 Inverse-Compton Gamma-ray Emission 8

3 Spectral Models **11**

3.1 **sync** 11

3.2 **invc** 12

3.3 **ntbrem** 13

3.4 **pizero** 14

4 Built-In Particle Distribution Functions **17**

4.1 **default** 17

4.2 **cbreak** 18

4.3 **ke_cutoff** 18

4.4 **etot** 19

4.5 **dermer** 19

4.6 **mori** 19

4.7 **boltz** 20

4.8 **rboltz** 20

4.9 **full1** 20

5 User-Defined Particle Distribution Functions **23**

6 Utility Functions **25**

6.1 **add_pdf** 25

6.2 **make_sync_table** 25

6.3 make_invc_table . . . 26
6.4 make_ntbrem_table . . . 26
6.5 ntb_set_process_weights . . . 27
6.6 particle_info_struct . . . 27
6.7 find_momentum_min . . . 28
6.8 nontherm_density . . . 28
6.9 nontherm_energy_density . . . 29
6.10 force_charge_conservation . . . 29
6.11 _sync_angular_integral . . . 29
6.12 _invc_photon_integral . . . 30
6.13 _ee_haug1 . . . 30
6.14 _ee_haug1_lab . . . 31
6.15 _ep_heitler1 . . . 31

Chapter 1

Introduction

As part of our work on cosmic-ray acceleration in supernova remnants, we wanted to search for evidence of curvature in relativistic particle momentum distributions. However, to the best of our knowledge, no publically-available spectral-fitting package included models for photon emission generated by relativistic particles (via synchrotron, inverse Compton, nonthermal bremsstrahlung and neutral-pion decay processes) and also allowed one to easily investigate the effect of different particle distribution functions.

XSPEC includes the synchrotron models SRCUT and SRESC, but these models are based on a fixed particle distribution function parameterization.

By implementing the necessary spectral models in a form accessible to the ISIS spectral analysis system, we gain the ability to fit these models not just to X-ray data, but also to radio and gamma-ray observations as well. By simultaneously fitting observations from several wave-bands, one can try to better constrain the important fit parameters.

This manual describes each of the spectral models and particle distribution functions and also provides a few simple usage examples. The physical assumptions used to create the models and various computational details are discussed in Houck & Allen (2006) [ApJS, 167, 26].

Chapter 2

Examples

In the following examples, we assume that the nonthermal module has been installed in the default location so that **isis** will find it automatically.

2.1 X-ray Synchrotron Emission

In this section, we describe how to fit X-ray data using the synchrotron model. For a more general discussion of how to use **isis** in data analysis, see the *ISIS documentation* http://space.mit.edu/cxc/isis/manual.html .

First, import the nonthermal module using

```
require ("nonthermal");
```

to make the spectral models and related functions available to **isis**.

Next, load the spectral data, background spectrum and instrument responses:

```
variable pha = load_data ("pha.fits");
assign_arf (load_arf ("arf.fits"), pha);
assign_rmf (load_rmf ("rmf.fits"), pha);
() = define_bgd (pha, "back.fits");
```

At this point, it may be desirable to rebin the data to improve the count-statistics in certain regions and to ignore selected spectral intervals. For example, to ensure that each bin contains at least 25 counts, use

```
rebin_data (pha, 25);
```

and, to fit only data in the range 2-8 keV, use

```
xnotice_en (pha, 2, 8);
```

Next, define the spectral model. For this example, we will use a simple model consisting of a single synchrotron emission component modified by line of sight absorption. To compute the synchrotron

emission spectrum, it is necessary to specify the underlying particle momentum distribution function. For this example, we will use the `default` momentum distribution function described in Houck & Allen (2006) [ApJS, 167, 26]. To model the effects of line of sight absorption, we will use the `phabs` function from XSPEC. The spectral model is then:

```
fit_fun ("phabs(1) * sync(1, default(1))");
```

Before fitting the data, is usually a good idea to choose initial parameter values such that the starting model is reasonably close to the data.

Once we're satisfied with the initial parameter values, perform the fit and then overplot the model and data histograms:

```
() = fit_counts;
rplot_counts (pha);
```

At this point, it is usually a good idea to investigate the neighborhood of the fitted parameters to make sure that the best possible fit has been achieved. For example, we might use `conf` to compute single parameter confidence limits for fit-parameters of interest.

2.2 Inverse-Compton Gamma-ray Emission

The process of fitting gamma-ray data is essentially identical to that used to fit X-ray data in the previous example. The primary difference is that, before the data can be loaded into isis, TeV gamma-ray spectra must first be cast into a suitable form.

One suitable form is an `ASCII` file with spectral data in four columns and with a small number of scalar parameters specified using header keywords. Using HEGRA observations of Cas A as a concrete example, we might create a file that looks like this:

```
# flux in photons/s/cm^2/bin
; object   Cas A
; bintype  counts
; xunit    TeV
; exposure 1.0
#  E_lo            E_hi            flux            error
   4.997493e-01    7.913994e-01    2.839012e-12    1.783329e-12
   7.913994e-01    1.256344e+00    3.729176e-13    2.399393e-13
   1.256344e+00    1.992807e+00    1.503555e-13    7.308948e-14
   1.992807e+00    3.168774e+00    7.704922e-14    3.311765e-14
   3.168774e+00    4.997493e+00    4.837415e-14    1.612472e-14
   4.997493e+00    7.913994e+00    1.115603e-14    1.247306e-14
   7.913994e+00    1.258409e+01    1.821113e-14    1.224027e-14
   1.258409e+01    2.001003e+01    1.804125e-15    3.373713e-15
```

Details of this format are described in the

ISIS documentation . Lines beginning with a '`#`' symbol are ignored and may be used to insert comments. The first two columns define the lower and upper edges of histogram bins. The bins must be in monotonic increasing order and may not overlap. The '`xunit`' keyword specifies the

bin coordinate units; in this case, the bin coordinates give the photon energy in TeV. The next two columns contain the flux in `photons/s/cm^2` and the associated uncertainty, assumed to be symmetric. Lines beginning with a semicolon (;) define various keywords recognized by **isis**. The `exposure` keyword specifies a nominal exposure time of one second.

Note that the '`bintype`' keyword labels the data as "counts" even though the spectral values are given in flux units. This subterfuge allows us to simultaneously fit X-ray data (in counts) and gamma-ray data (in flux units). However, to make this work, we must remember to turn off one of the normal data-input validation tests. By default, when **isis** reads counts spectra, it requires that the uncertainty in the number of counts be greater than or equal to one. If the input data do not satisfy this requirement, **isis** replaces the relevant input uncertainties with acceptable values. To keep **isis** from modifying our input uncertainties in this way, we set the intrinsic variable `Minimum_Stat_Err` to a small positive value (smaller than any of the input uncertainties).

Having converted the Cas A HEGRA spectrum into the above format, we can load the data into isis:

```
Minimum_Stat_Err = 1.e-30;
variable id = load_data ("casa_hegra_data.txt");
```

Once the data have been loaded, we can define and fit a model exactly as before. For example, to fit a model which includes both inverse Compton and neutral-pion decay components, we can use

```
fit_fun ("invc(1, default(1)) + pizero(1, default(2))");
```

Note that two particle distribution functions appear in this spectral model. The inverse Compton component uses `default(1)` to define the electron momentum distribution and the neutral-pion decay component uses `default(2)` to define the proton momentum distribution.

Chapter 3

Spectral Models

Note on differential vs. histogram forms

Differential and histogram forms of each spectral model are provided. The histogram form is the one used in fitting models to data. For this reason, input data must be converted to histogram form. The differential form is normally used for testing and for making plots. See the *ISIS documentation* for additional details.

3.1 sync

Synopsis

Synchrotron spectrum model

Usage

```
sync(Int_Type i, String_Type pdf_name, Double_Type pdf_params[])
```

Description

The `sync` function computes the synchrotron spectrum for a specified nonthermal particle distribution function. The synchrotron spectrum parameters are

```
B_tot = total magnetic field strength [microgauss].
 norm = normalization = A_e V / (4 \pi d^2)
```

where

```
  A_e = nonthermal electron density at 1 GeV [cm^-3 GeV^-1]
    V = emitting volume [cm^3]
    d = distance [cm]
```

The first argument of the `sync` function is an integer index identifying a particular instance of this fit function. The second and third arguments specify the particle distribution function (PDF) and its parameters.

Normally, the second and third arguments are provided as return values by a special fit-function that acts as an interface to the desired PDF. For example,

```
fit_fun ("sync(1, default(1))")
```

causes the synchrotron function calculation to use the PDF named `default`, which has parameters

```
    index = momentum power-law index
curvature = amount of curvature above 1 GeV
 E_cutoff = cut-off momentum [TeV].
```

When this fit-function is evaluated, `default(1)` returns second and third parameters needed by `sync`.

By default, the value of the angular integral is determined via spline interpolation in a precomputed table. To compute the angular integral value via direct integration, set the intrinsic variable `_sync_interpolate` to a non-zero value.

See Also

6.11 (_sync_angular_integral)

3.2 invc

Synopsis

Inverse Compton spectrum model

Usage

```
invc(Int_Type i, String_Type pdf_name, Double_Type pdf_params[])
```

Description

The `invc` function computes the inverse Compton spectrum for a specified nonthermal particle distribution function. The inverse Compton spectrum parameters are

```
T_photon = radiation field temperature [K]
    norm = normalization = A_e V / (4 \pi d^2)
```

where

```
A_e = nonthermal electron density at 1 GeV [cm^-3 GeV^-1]
  V = emitting volume [cm^3]
  d = distance [cm]
```

The first argument of the `invc` function is an integer index identifying a particular instance of this fit function. The second and third arguments specify the particle distribution function (PDF) and its parameters.

Normally, the second and third arguments are provided as return values by a special fit-function that acts as an interface to the desired PDF. For example,

```
fit_fun ("invc(1, default(1))")
```

causes the inverse Compton function calculation to use the PDF named `default`, which has parameters

```
    index = momentum power-law index
curvature = amount of curvature above 1 GeV
 E_cutoff = cut-off momentum [TeV].
```

When this fit-function is evaluated, `default(1)` returns second and third parameters needed by `invc`.

By default, the value of the integral over incident photon energies is determined via spline interpolation in a pre-computed table and the value of the `T_photon` parameter is not used. To compute this integral via direct integration, set the intrinsic variable `_invc_interpolate` to a non-zero value.

See Also

6.12 (_invc_photon_integral)

3.3 ntbrem

Synopsis

Nonthermal bremsstrahlung spectrum model

Usage

```
ntbrem(Int_Type i, String_Type pdf_name, Double_Type pdf_params[])
```

Description

The `ntbrem` function computes the nonthermal bremsstrahlung spectrum for a specified nonthermal particle distribution function, assuming a stationary target. The primary nonthermal bremsstrahlung spectrum parameter is the normalization,

```
norm = n_t A_e V / (4 \pi d^2)
```

where

```
n_t = density of target nuclei [cm^-3]
A_e = nonthermal electron density at 1 GeV [cm^-3 GeV^-1]
  V = emitting volume [cm^3]
  d = distance [cm]
```

The first argument of the `ntbrem` function is an integer index identifying a particular instance of this fit function. The second and third arguments specify the particle distribution function (PDF) and its parameters.

Normally, the second and third arguments are provided as return values by a special fit-function that acts as an interface to the desired PDF. For example,

```
fit_fun ("ntbrem(1, default(1))")
```

causes the nonthermal bremsstrahlung function calculation to use the PDF named `default`, which has parameters

```
    index = momentum power-law index
curvature = amount of curvature above 1 GeV
 E_cutoff = cut-off momentum [TeV].
```

When this fit-function is evaluated, `default(1)` returns second and third parameters needed by `ntbrem`.

By default, the value of the electron-electron differential cross-section is determined via spline interpolation in a pre-computed table. To compute these cross-sections without interpolation, set the intrinsic variable `_ntbrem_interpolate` to a non-zero value.

The relative contributions of electron-electron and electron-proton bremsstrahlung are controlled by the `ntb_set_process_weights` function.

To use an alternate, somewhat faster, numerical integration algorithm, set the integer intrinsic variable `_ntbrem_integration_method` to a non-zero value. Note that changing the numerical integration method will cause small changes in the computed model values.

See Also

6.5 (ntb_set_process_weights), 6.13 (_ee_haug1), 6.14 (_ee_haug1_lab)

3.4 pizero

Synopsis

Neutral-pion decay spectrum model

Usage

```
pizero(Int_Type i, String_Type pdf_name, Double_Type pdf_params[])
```

Description

The `pizero` function computes the neutral-pion decay gamma-ray spectrum for a specified nonthermal particle distribution function. The primary pion decay spectrum parameter is the normalization,

```
norm = n_p A_p V / (4 \pi d^2)
```

where

```
n_p = density of target protons [cm^-3]
A_p = nonthermal proton density at 1 GeV [cm^-3 GeV^-1]
  V = emitting volume [cm^3]
  d = distance [cm]
```

The first argument of the `pizero` function is an integer index identifying a particular instance of this fit function. The second and third arguments specify the particle distribution function (PDF) and its parameters.

Normally, the second and third arguments are provided as return values by a special fit-function that acts as an interface to the desired PDF. For example,

```
fit_fun ("pizero(1, default(1))")
```

causes the neutral-pion decay function calculation to use the PDF named `default`, which has parameters

```
    index = momentum power-law index
curvature = amount of curvature above 1 GeV
 E_cutoff = cut-off momentum [TeV].
```

When this fit-function is evaluated, `default(1)` returns second and third parameters needed by `pizero`.

See Also

Chapter 4

Built-In Particle Distribution Functions

4.1 default

Synopsis

default PDF

Usage

```
default(id)
```

Description

This distribution function provides a power-law in momentum with curvature above some specific particle momentum and with an exponential cut-off.

The distribution function has the form

```
N(p) = A (p/p0)^f(p,a) exp((p0-p)/cutoff)
```

where

```
f(p,a) = -index + curvature * log10(p/p0),  p>p0
f(p,a) = -index,                            p<=p0
```

The value of `p0` is determined by the value of `Min_Curvature_Pc`=1 GeV by default.

The fit parameters are

```
index       power-law index
curvature   curvature parameter
cutoff      cut-off energy [TeV]
```

See Also

6.1 (add_pdf)

4.2 cbreak

Synopsis

cooling-break PDF

Usage

```
cbreak(id)
```

Description

This distribution function is the same as the default distribution function except that it also has a "cooling break" above a particular momentum. The cooling break is implemented by adding an additional multiplicative factor of the form

```
        (break/pc), for pc > break
g(p) =
        1,          for pc < break
```

where `p` is the particle momentum and `break` is the 'break energy'.

The fit parameters are then

```
index       power-law index
curvature   curvature parameter
cutoff      cut-off energy [TeV]
break       cooling-break energy [TeV]
```

See Also

6.1 (add_pdf)

4.3 ke_cutoff

Synopsis

PDF with cutoff in particle kinetic energy

Usage

```
ke_cutoff(id)
```

Description

This distribution function is the same as the default distribution function except that the exponential cutoff depends on kinetic energy, `T`, instead of momentum, `p`.

See Also

6.1 (add_pdf)

4.4 etot

Synopsis

Energy-dependent PDF

Usage

```
etot(id)
```

Description

This distribution function is a power-law in total-energy,

```
N(p) = A (E/E0)^(-index)
```

where `E0`=1 GeV.

The power-law index is the only fit-parameter.

See Also

6.1 (add_pdf)

4.5 dermer

Synopsis

PDF from Dermer (1986)

Usage

```
dermer(id)
```

Description

This distribution function is a power-law in total-energy, with momentum-dependent curvature above the momentum specified by the intrinsic variable `Min_Curvature_Pc`, which is 1 GeV by default.

The fit parameters are then

```
index       power-law index
curvature   curvature parameter
```

See Also

6.1 (add_pdf)

4.6 mori

Synopsis

PDF from Mori (1997)

Usage

```
mori(id)
```

Description

This distribution function was taken from Mori (1997) and is designed to represent the Galactic cosmic-ray proton distribution. It has no free parameters.

See Also

6.1 (add_pdf)

4.7 boltz

Synopsis

Non-relativistic Maxwell-Boltzmann Distribution

Usage

```
boltz(id)
```

Description

This distribution function provides a non-relativistic Maxwell-Boltzmann thermal particle distribution. The only fit parameter is the temperature, `kT`, in keV.

See Also

6.1 (add_pdf)

4.8 rboltz

Synopsis

Relativistic Maxwell-Boltzmann Distribution

Usage

```
rboltz(id)
```

Description

This distribution function provides a relativistic Maxwell-Boltzmann thermal particle distribution. The only fit parameter is the temperature, `kT`, in keV.

See Also

6.1 (add_pdf), 4.9 (full1)

4.9 full1

Synopsis

RMB + Cutoff Power-Law PDF

Usage

```
full1(id)
```

Description

This particle distribution function consists of a relativistic Maxwell-Boltzmann thermal distribution (`rboltz`) plus a nonthermal tail (`default`). The fit parameters are those of `rboltz` plus `default` plus an additional parameter, `A_GeV` which gives the density of nonthermal particles with p=1 GeV/c [cm^-3 GeV^-1].

See Also

6.1 (add_pdf), 4.8 (rboltz), 4.1 (default)

Chapter 5

User-Defined Particle Distribution Functions

A user-defined particle distribution function may be implemented in a compiled language such as C or Fortran, compiled as a shared library, and then imported into **isis** at run-time via dynamic linking.

A detailed example of how to do this is provided in the '`examples`' subdirectory of the module source-code distribution.

Chapter 6

Utility Functions

6.1 add_pdf

Synopsis

Add a particle distribution function

Usage

```
add_pdf (String_Type libname, String_Type pdfname [, param_names[],
param_values[], freeze[], min[], max[]])
```

Description

The `add_pdf` function can be used to load a particle distribution function (PDF) from a shared library. An example implementation of such an object is provided in the `examples` directory of the source code distribution.

The first two arguments give the name of the shared library and the name of the PDF. If the PDF has any fit parameters, the remaining array arguments should provide the parameter names, default values, default freeze/thaw state and default allowed minimum and maximum values.

See Also

6.2 make_sync_table

Synopsis

Generate a synchrotron lookup table

Usage

```
make_sync_table ([String_Type file])
```

Description

The `make_sync_table` function can be used to generate a new lookup table for the angular integration that arises in the computation of the `sync` model.

The file name may be provided as a parameter. If no file name is provided, the default file name is used. The output file is a FITS bintable.

To customize details of the table computation, modify the script `lib/sync_make_table.sl` which is installed in `${prefix}/share/isis/nonthermal`.

See Also

6.11 (_sync_angular_integral)

6.3 make_invc_table

Synopsis

Generate an inverse Compton lookup table

Usage

```
make_invc_table ([String_Type file])
```

Description

The `make_invc_table` function can be used to generate a new lookup table for the integral over incident photons that arises in the computation of the `invc` model.

The file name may be provided as a parameter. If no file name is provided, the default file name is used. The output file is a FITS bintable.

To customize details of the table computation, modify the script `lib/invc_make_table.sl` which is installed in `${prefix}/share/isis/nonthermal`.

A blackbody radiation field is used by default. To use a different radiation field, it is necessary to modify the source code [replacing src/bbody.c and making any other necessary changes]

See Also

6.12 (_invc_photon_integral)

6.4 make_ntbrem_table

Synopsis

Generate an ee Bremsstrahlung lookup table

Usage

```
make_ntbrem_table ([String_Type file])
```

Description

The `make_ntbrem_table` function can be used to generate a new lookup table for the electron-electron bremsstrahlung differential cross-sections that arise in the computation of the `ntbrem` model.

The file name may be provided as a parameter. If no file name is provided, the default file name is used. The output file is a FITS bintable.

To customize details of the table computation, modify the script `lib/ntbrem_make_table.sl` which is installed in `${prefix}/share/isis/nonthermal`.

See Also

6.13 (_ee_haug1), 6.14 (_ee_haug1_lab)

6.5 ntb_set_process_weights

Synopsis

Set the relative weights of ee and ep bremsstrahlung

Usage

```
ntb_set_process_weights (wt_ee, wt_ep)
```

Description

The `ntb_set_process_weights` function can be used to set the weights used for electron-electron and electron-proton bremsstrahlung. The total contribution from both processes is

```
S(E) = wt_ee * S_ee(E) + wt_ep * S_ep(E)
```

The fluxes are scaled relative to the ambient number density of nuclei, `n`. The default weights are appropriate for a fully ionized gas with cosmic abundances so that most electrons come from hydrogen and helium. In that case, the number fractions of hydrogen and helium are roughly

```
X_H  = 1.0/1.1 = 0.9091,
X_HE = 0.1/1.1 = 0.0909.
```

The electron-electron bremsstrahlung contribution, proportional to the ambient electron density has weight

```
wt_ee = X_H + 2*X_HE = 1.09091
```

and the total electron-proton bremsstrahlung contribution, proportional to `n*Z^2`, has weight

```
wt_ep = X_H + 4*X_HE = 1.27273
```

See Also

6.6 particle_info_struct

Synopsis

Get a template structure for specifying the PDF

Usage

```
Struct_Type = particle_info_struct ()
```

Description

The `particle_info_struct` returns a template structure with the following fields:

```
__Name__        __Definition__
pdf_name         name of the particle distribution function
params           parameter values
particle_type    electron=0, proton=1
n_GeV            density at 1 GeV [cm^-3 GeV^-1]
kT               temperature of thermal particles
n_th             density of thermal particles
```

See Also

6.7 find_momentum_min

Synopsis

Find the momentum at which the thermal and nonthermal PDFs intersect

Usage

```
pc_min = find_momentum_min (Struct_Type p)
```

Description

The `find_momentum_min` function takes a structure defining the thermal and nonthermal components of the particle distribution function and computes the momentum at which those two distributions contribute equally.

If the thermal PDF is everywhere below the nonthermal PDF, the momentum at the thermal peak is returned.

See Also

6.6 (particle_info_struct)

6.8 nontherm_density

Synopsis

Compute the density of nonthermal particles

Usage

```
n = nontherm_density (Struct_Type p)
```

Description

The `nontherm_density` function takes a structure defining the thermal and nonthermal components of the particle distribution function (PDF) and computes the integral over the nonthermal PDF. The lower limit of this integral is the momentum at which the thermal and nonthermal PDFs intersect. If the thermal PDF is everywhere below the nonthermal PDF, the momentum at the thermal peak is used.

See Also

6.7 (find_momentum_min)

6.9 nontherm_energy_density

Synopsis

Compute the energy density of nonthermal particles

Usage

```
n = nontherm_energy_density (Struct_Type p)
```

Description

The `nontherm_energy_density` function takes a structure defining the thermal and nonthermal components of the particle distribution function (PDF) and computes the energy density in nonthermal particles by integrating over the nonthermal PDF. The lower limit of this integral is the momentum at which the thermal and nonthermal PDFs intersect. If the thermal PDF is everywhere below the nonthermal PDF, the momentum at the thermal peak is used.

See Also

6.7 (`find_momentum_min`)

6.10 force_charge_conservation

Synopsis

Enforce charge conservation by adjusting the proton norm

Usage

```
force_charge_conservation (electrons, protons [, method])
```

Description

The `force_charge_conservation` function take structures defining the electron and proton particle distribution functions and adjusts the normalization of the nonthermal proton distribution (protons.n_GeV) to enforce some definition of charge conservation. The supported definitions are

```
method=0 equal density of electrons and protons at some chosen
         injection kinetic energy
method=1 equal integrated density of nonthermal electrons and protons
```

The default is `method=0`.

See Also

6.6 (`particle_info_struct`)

6.11 _sync_angular_integral

Synopsis

Synchrotron angular integral

Usage

```
_sync_angular_integral (x, interp)
```

Description

The `_sync_angular_integral` function computes the angular integral needed for the synchrotron spectrum model. The first argument, `x`, is defined as

```
x = f / f_c
```

where `f_c` is the synchrotron critical frequency.

If the second argument is non-zero, the integral will be evaluated by spline interpolation on a pre-computed table. Otherwise, the value will be computed by numerical integration.

See Also

6.12 _invc_photon_integral

Synopsis

Inverse Compton integral over incident photon energies

Usage

```
_invc_photon_integral (gamma, omega, T, interp)
```

Description

The `_invc_photon_integral` function computes the integral over incident photon energies needed for the inverse Compton spectrum model. The first two parameters are

```
gamma = 1/sqrt (1 - beta^2)
omega = E / (m c^2)
```

where gamma is the Lorentz factor of the electron, E is the energy of the incident photon and m is the electron mass. The third argument is the radiation field temperature [K].

If the fourth is non-zero, the integral will be evaluated by spline interpolation on a pre-computed table. Otherwise, the value will be computed by numerical integration.

See Also

6.13 _ee_haug1

Synopsis

Electron-electron bremsstrahlung lab frame differential cross-section

Usage

```
sigma = _ee_haug1 (T, omega)
```

Description

The _ee_haug1 function computes the lab-frame differential cross-section for electron-electron bremsstrahlung by applying a numerical Lorentz transformation to the center of momentum frame cross-section. The parameters are

```
    T = incident electron kinetic energy in units of the
        electron rest energy.
omega = scattered photon energy in units of the electron
        rest energy
```

The cross-section is in units cm^2/erg.

See Also

6.14 (_ee_haug1_lab)

6.14 _ee_haug1_lab

Synopsis

Electron-electron bremsstrahlung lab frame differential cross-section

Usage

```
sigma = _ee_haug1_lab (T, omega)
```

Description

The _ee_haug1_lab function computes the lab-frame differential cross-section for electron-electron bremsstrahlung. The parameters are

```
    T = incident electron kinetic energy in units of the
        electron rest energy.
omega = scattered photon energy in units of the electron
        rest energy
```

The cross-section is in units cm^2/erg.

See Also

6.15 _ep_heitler1

Synopsis

Bethe-Heitler cross-section

Usage

```
sigma = _ep_heitler1 (T, omega)
```

Description

The _ep_heitler1 function computes the Bethe-Heitler cross-section for electron-ion bremsstrahlung. The parameters are

```
      T = incident electron kinetic energy in units of the
          electron rest energy.
  omega = scattered photon energy in units of the electron
          rest energy
```

The cross-section is in units cm^2/erg.

See Also